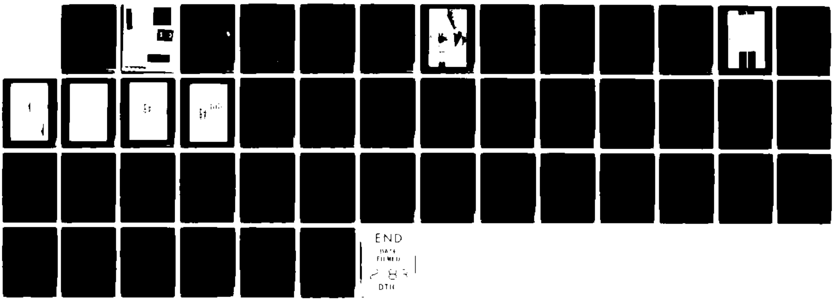
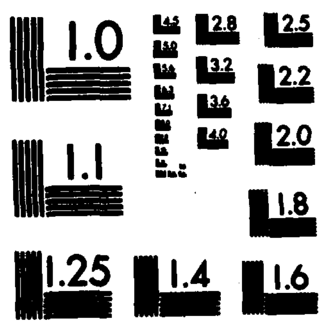
AD-A122 915 SPATIAL MANAGEMENT OF DATA(U) COMPUTER CORP OF AMERICA CAMBRIDGE MA C F HEROT MAR 79 MDA903-78-C-0122 1/1

UNCLASSIFIED F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

"Original contains color plates: All DTIC reproductions will be in black and white"

3/79

SPATIAL MANAGEMENT OF DATA

Christopher F. Herot

Computer Corporation of America
575 Technology Square
Cambridge, Massachusetts

March, 1979

The work reported herein was sponsored by the Defense Advanced Research Projects Agency under contract MDA903-78-C-0122.

AO 3487
50 C7

SPATIAL MANAGEMENT OF DATA

ABSTRACT

Spatial Data Management is a technique for organizing and retrieving information by positioning it in a Graphical Data Space (GDS). This Graphical Data Space is viewed through a color raster scan display which enables users to traverse the GDS surface or zoom into the image to obtain greater detail. In contrast to conventional database management systems in which users access data by asking questions in a formal query language, a Spatial Data Management System (SDMS) presents the information graphically in a form which seems to encourage browsing and to require less prior knowledge of the contents and organization of the database.

This paper presents an overview of the SDMS concept and describes its implementation in a prototype system for retrieving information from both a symbolic database management system and from an optical videodisk.

1-1

Table of Contents

1. INTRODUCTION

Spatial Data Management is motivated by the needs of a growing community of people who need to access information in a database management system (DBMS) but who are not trained in the use of such systems. The information in an SDMS is expressed in graphical form and presented in a spatial framework, so that the information is more accessible and its structure is more obvious than in a conventional DBMS. In this way, a user can find the information he seeks without having to specify it precisely or know exactly where in the DBMS it is stored.

The graphical data space is accessed through a set of three color, raster-scan displays as illustrated in Figure 1. The left-most of the three screens presents a "world-view" map of the entire data surface. A magnified portion of this data surface is simultaneously displayed on the main screen in the center. The location on the data surface of this magnified portion is indicated by a highlighted rectangle which appears on the world-view map. The user can control which portion of the data surface appears on the main display by pressing on the joy stick shown in the foreground of the figure in the user's left hand. Pressing the joy stick in any given direction causes the user's magnified window to move in that direc-

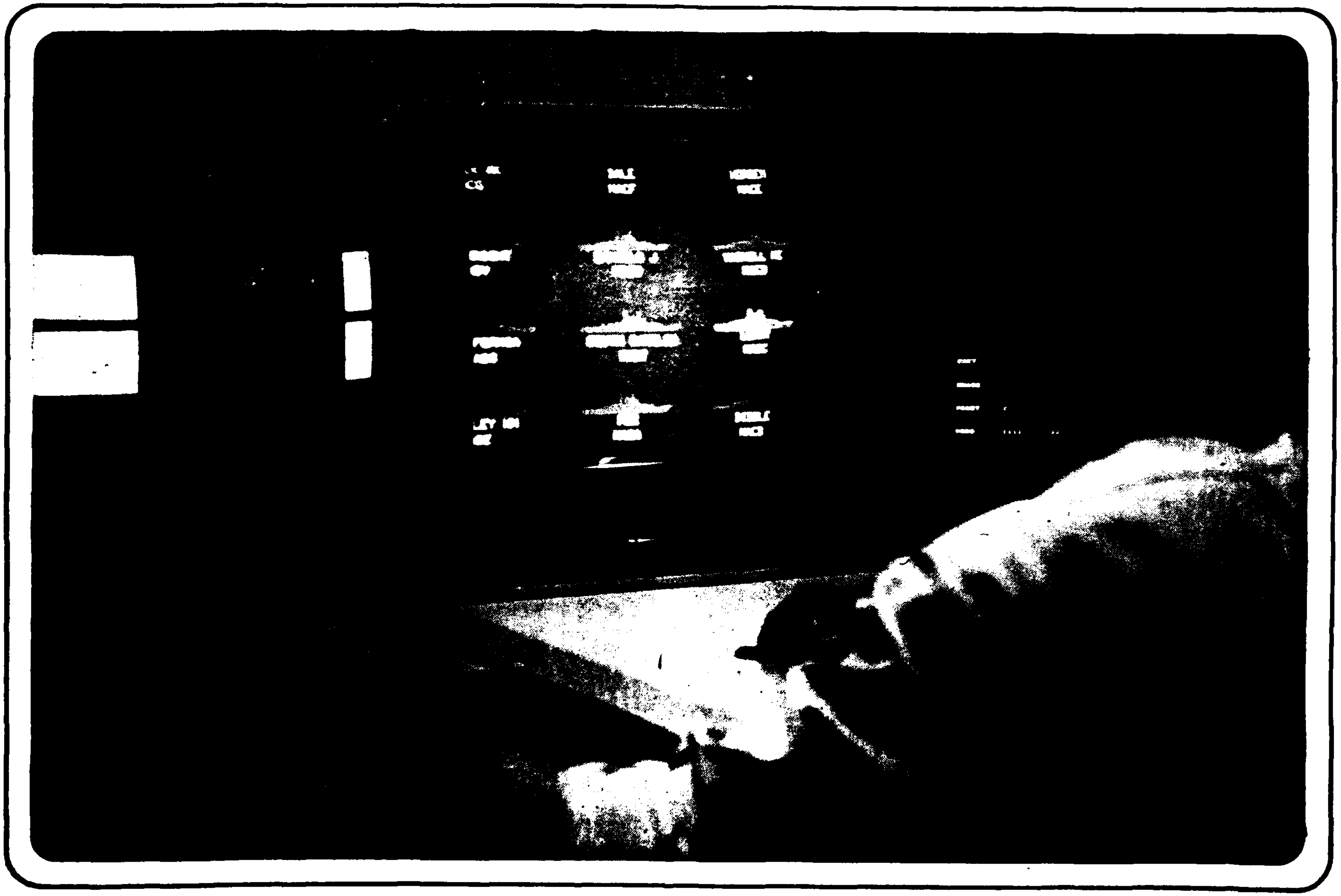


Figure 1. User Station

tion over the data surface. This motion is reflected in the corresponding motion of the highlighted rectangle on the world-view map.

The data presented to the user on the main display can come from a variety of sources. The three sources described in this paper are:

1. images stored as bit-arrays on a digital disk,
2. a symbolic database management system, and
3. an optical videodisk.

The use of the system is best illustrated through an extended example, given in Section 2.

Section 3 contrasts SDMS with conventional database management systems.

Section 4 details the use of the optical videodisk for storage and retrieval of analogue video information.

Section 5 describes some additional features of the system for defining the relationship between graphical and symbolic representations of data.

Finally, Section 6 gives a brief account of the progress on the implementation of the SDMS prototype.

2. EXAMPLE

This section illustrates the use of SDMS as an interface to a symbolic database management system.

The user in this example is examining a database of ships. This database originated as a conventional database managed by the INGRES database management system [HELD, STONEBRAKER, WONG]. A fragment of the SHIP relation in the DBMS is shown in table 1.

The procedure through which the database administrator produced the graphical representation used in the example is described in Section 2.2.

uic	nam	type	nat	ircs	beam	ready
N00001	CONSTELLATION	CV	US	NABC	130	1
N00002	KENNEDY JF	CV	US	NABD	130	1
N00003	KITTY HAWK	CV	US	NABE	130	2
N00004	AMERICA	CV	US	NABF	130	5
N00005	SARATOGA	CV	US	NABG	130	1
N00006	INDEPENDENCE	CV	US	NABH	130	1
N00007	LOS ANGELES	SSN	US	NABI	33	1
N00008	BATON ROUGE	SSN	US	NABJ	33	1
N00009	PHILADELPHIA	SSN	US	NABK	33	1
N00010	STURGEON	SSN	US	NABL	32	1
N00011	WHALE	SSN	US	NABM	32	1
N00012	TAUTOG	SSN	US	NABN	32	1
N00013	GRAYLING	SSN	US	NABO	32	1
N00014	POGY	SSN	US	NABP	32	1
N00015	ASPRO	SSN	US	NABQ	32	1
N00016	SUNFISH	SSN	US	NABR	32	1
N00017	CALIFORNIA	CGN	US	NABS	61	1
N00018	SOUTH CAROLINA	CGN	US	NABT	61	1
N00019	DANIELS J	CG	US	NABU	55	1
N00020	WAINWRIGHT	CG	US	NABV	55	1
N00021	JOUETT	CG	US	NABW	55	1
N00022	HORNE	CG	US	NABX	55	1
N00023	STERETT	CG	US	NABY	55	3
N00024	STANDLEY WH	CG	US	NABZ	55	1
N00025	FOX	CG	US	NACA	55	1
N00026	BIDDLE	CG	US	NACB	55	1
N00027	LEAHY	CG	US	NACC	55	4
N00028	YARNELL HE	CG	US	NACD	55	1
N00029	WORDEN	CG	US	NACE	55	1
N00030	DALE	CG	US	NACF	55	1
N00031	TURNER RK	CG	US	NACG	55	1
N00032	GRIDLEY	CG	US	NACH	55	1
N00033	ENGLAND	CG	US	NACI	55	1
N00034	HALSEY	CG	US	NACJ	55	1
N00035	REEVES	CG	US	NACK	55	3
N00036	ADAMS CF	DDG	US	N 47	1	
N00040	BARNEY	DDG	US	NACP	47	1
N00041	WILSON HB	DDG	US	NACQ	47	1
N00042	MCCORMICK L	DDG	US	NACR	47	1
N00043	TOWERS	DDG	US	NACS	47	1
N00044	SELLERS	DDG	US	NACT	47	1
N00045	ROBISON	DDG	US	NACU	47	1

Table 1: Ship database

2.1 Retrieval of Data

Figures 2 through 6 illustrate the use of SDMS by a user to retrieve information from the database.

Figure 2 shows the world-view map presented on the left most of the three screens. It is an all-encompassing view of the data laid out on the graphical data surface. The map has been divided into two rows, one for each of the two countries having ships in the database. Each row is further divided into columns, one for each class of ship. Within each column, each ship is represented by a colored shape, refered to as an icon.

The color of each icon indicates the ship's readiness, one of the attributes of each ship in the symbolic database. Green indicates the highest readiness, followed by yellow, orange, and red.

The size of each icon is a function of the beam of its associated ship. This is most apparent among the Russian carriers located in the lower left corner of the data surface.

In the blue section in the center of the top row, one can see a highlighted rectangle which indicates the portion of the data surface which is shown in the magnified view presented on the main display. The position of this

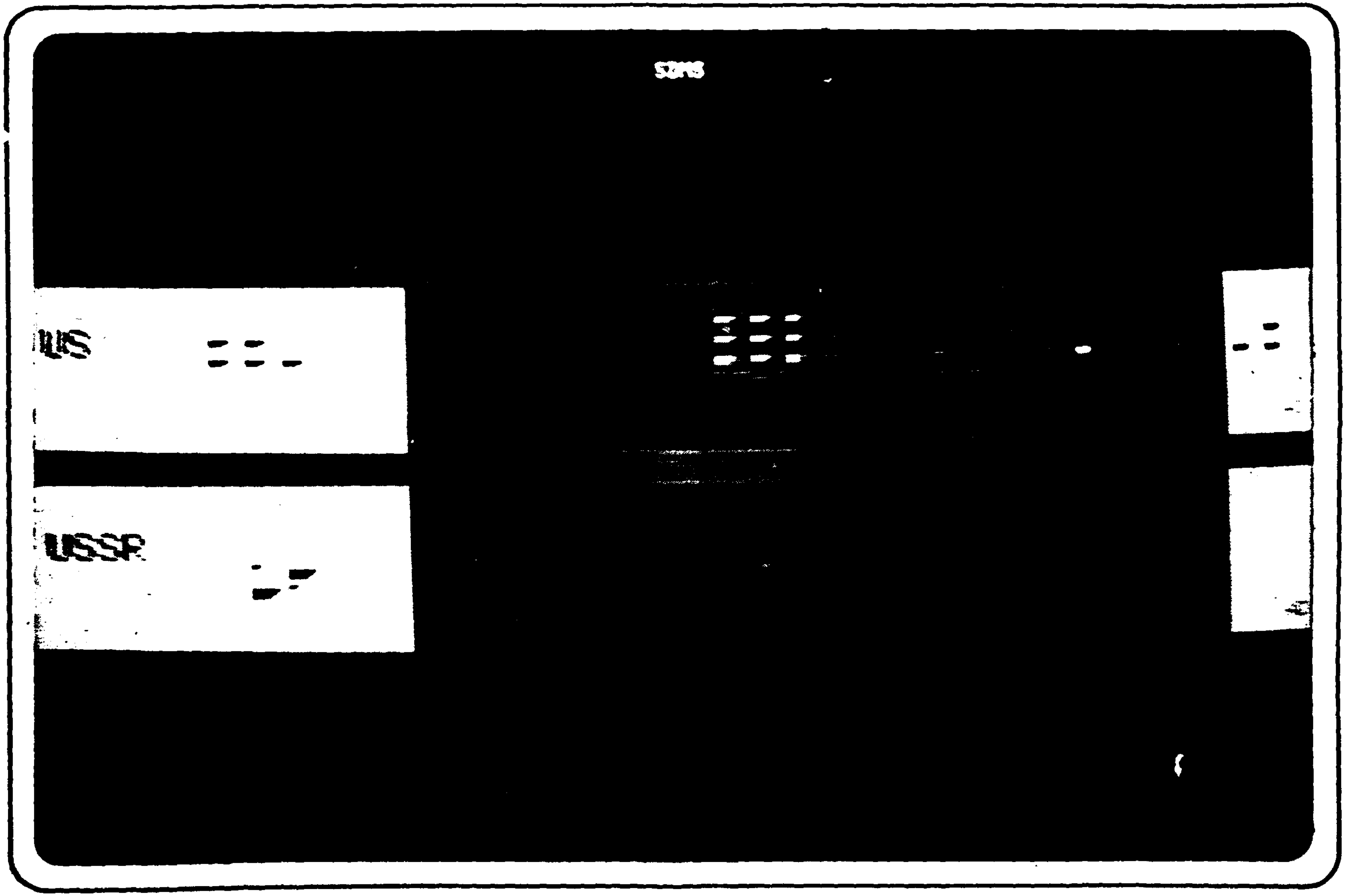


Figure 2. World-view map

rectangle, and thus the portion of the data surface shown on the main display, is controlled by the joy stick shown in figure 1.

Figure 3 shows the main data display which appears on the center screen. Notice that while the position and color of the ships is the same as in the highlighted rectangle of the world view map, the magnified view is more detailed. The icons have a more detailed shape, and each shape has two text strings beneath it which give the ship's name and international radio call sign.

Given this graphical view of the symbolic database, let us see how a user would go about retrieving information on a specific ship, in this case the "Horne". In order to do so, he pulls the joy stick down towards him, indicating that he wishes to move the magnified view down towards the bottom of the data surface. When the Horne is in the center of the screen, as shown in Figure 4, he releases the joy stick.

Now, in order to get a more detailed view, he twists the joy stick clockwise, instructing the system to expand the magnification of the picture shown on the screen. One instant in this zooming process is captured in figure 5.

Once the machine has magnified the image, it then adds more detail as shown in figure 6. The outline of the ship

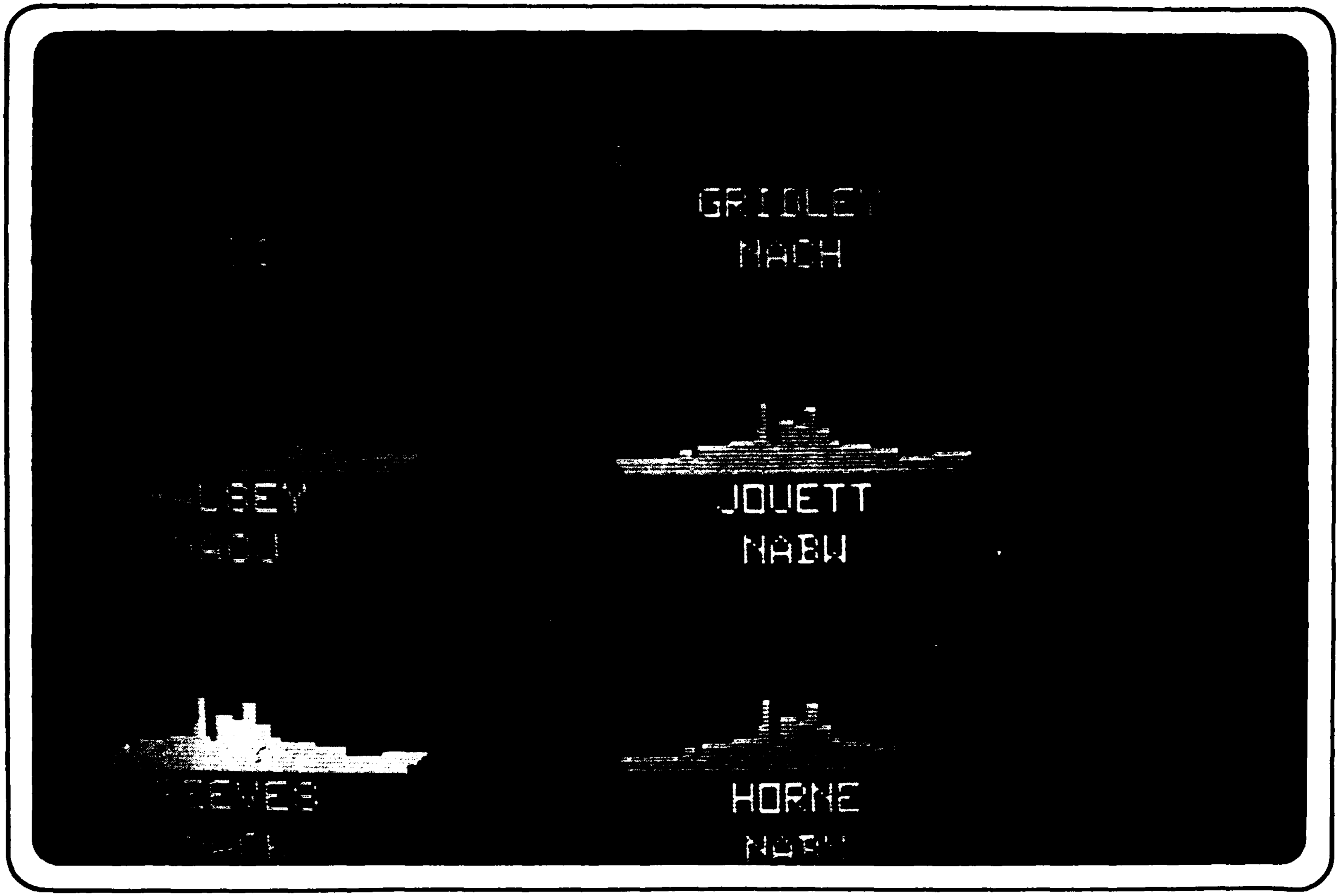


Figure 3. Magnified view of data surface

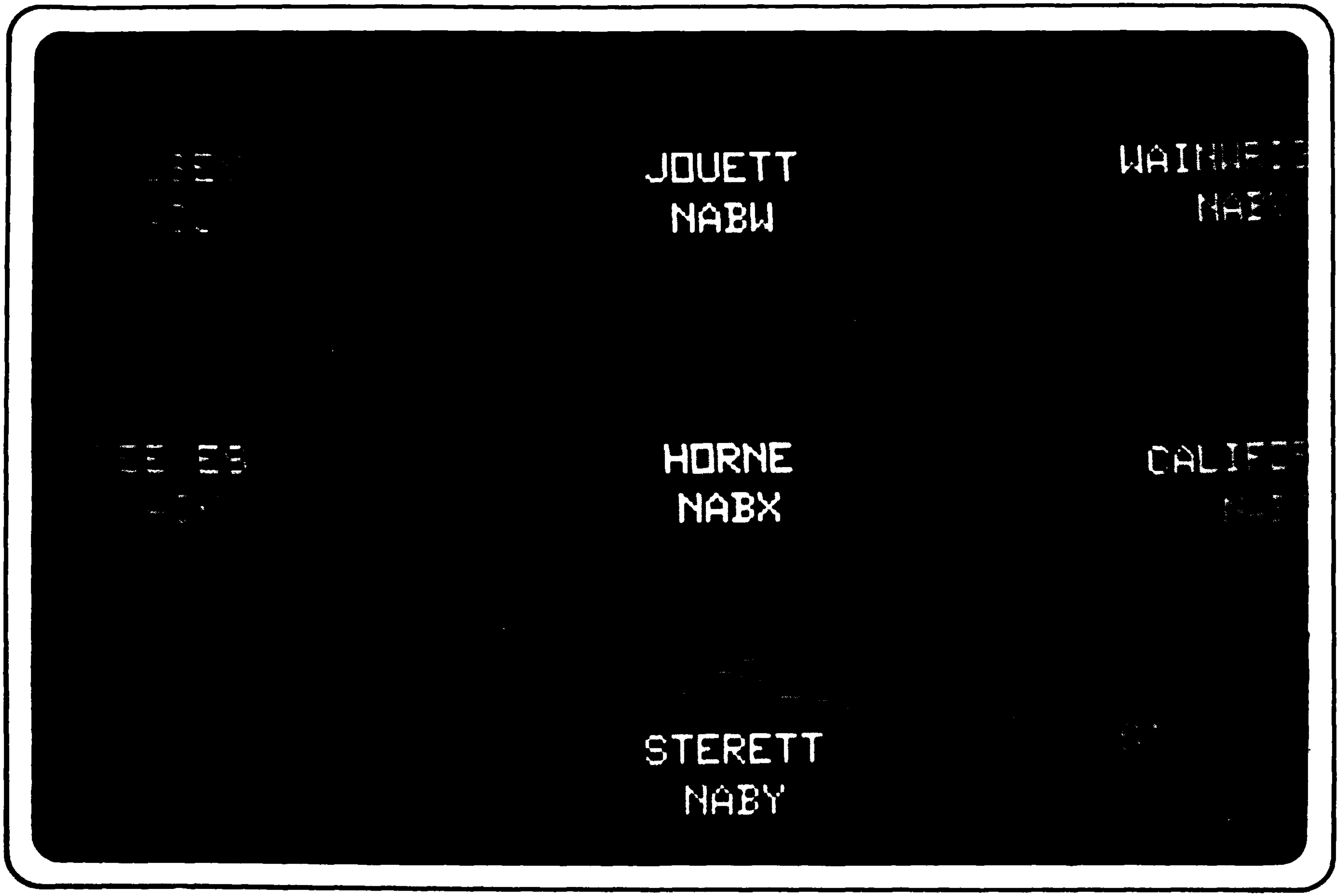


Figure 4. Magnified view with "Horne" centered

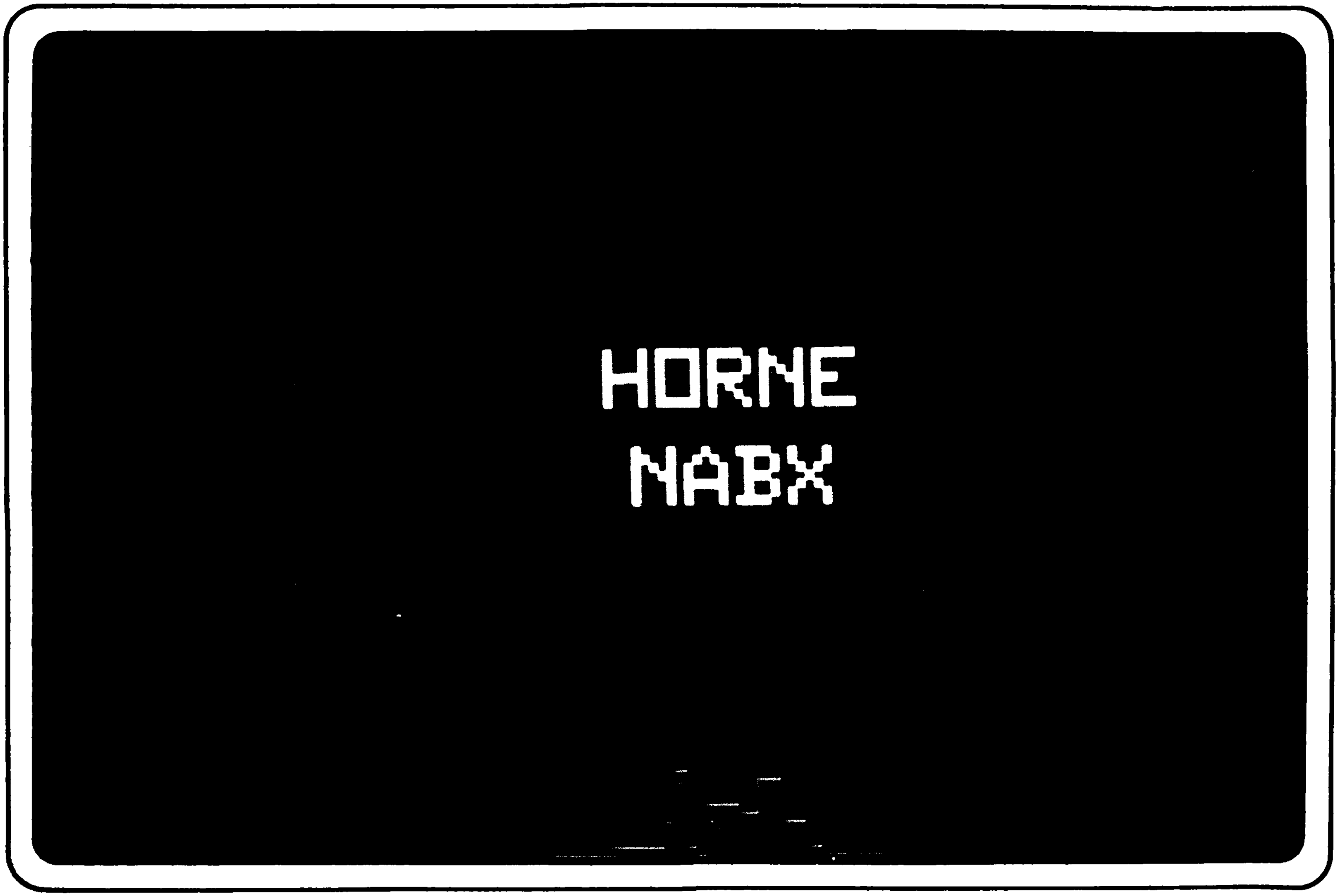


Figure 5. Zoomed in view of data surface

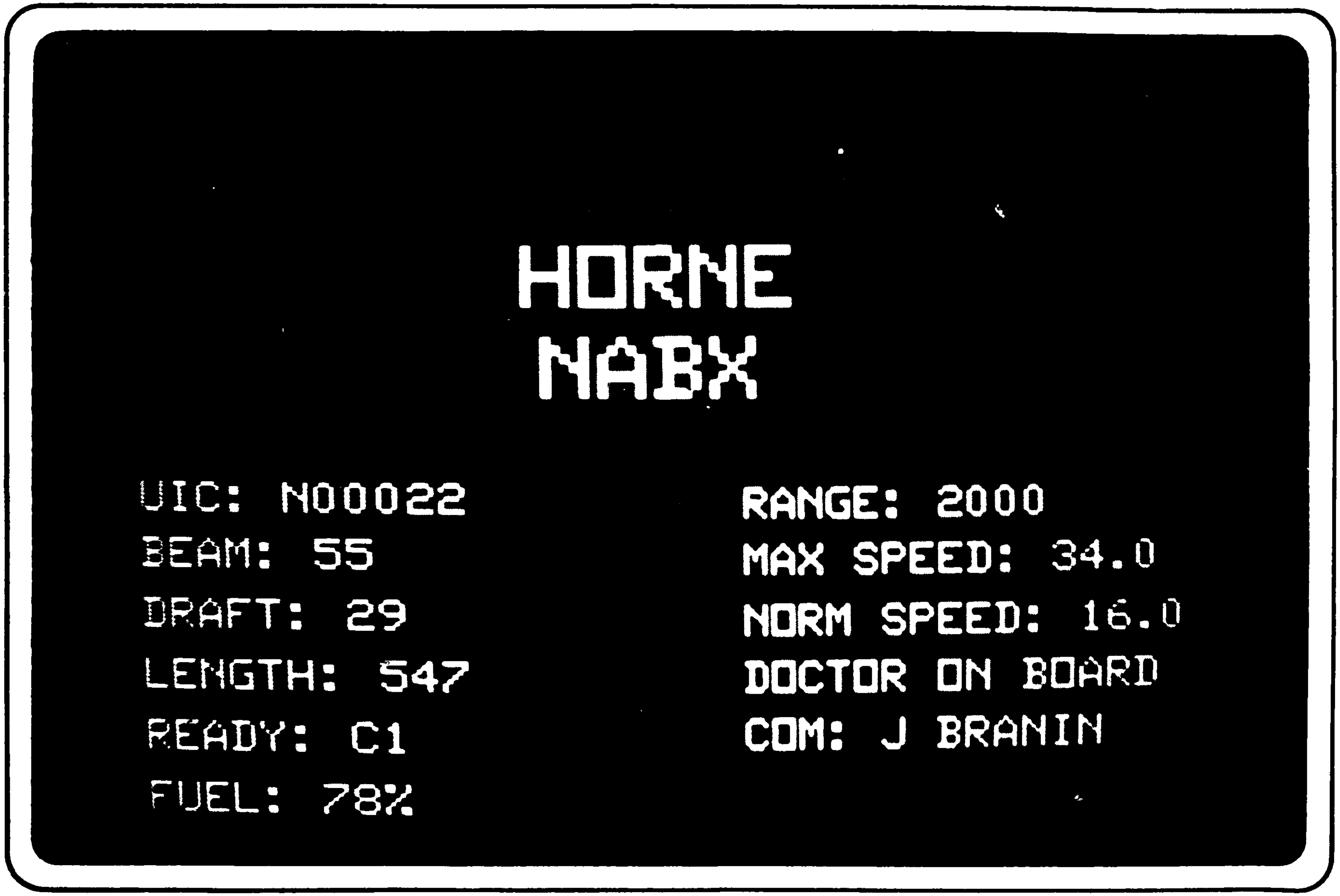


Figure 6. More detailed view of data surface

becomes more detailed, and the space under the ship is filled with values of attributes from the symbolic database.

At this point, the user has a choice of actions. He can move laterally across the data surface and examine similarly detailed views of adjacent ship icons. He can twist the joystick in the counterclockwise direction to return to the less detailed views of figures 4 and 5. Thirdly, if the database administrator had provided for it, the user could once again twist the joystick clockwise and magnify the view of Figure 6, resulting in a still more detailed icon with yet more attribute values displayed.

2.2 Creation of the Graphical Data Surface

The preceding paragraphs described how a user might use a graphical view of a symbolic database to retrieve some information. In order to construct such a view, the database administrator must first describe to the system how each icon should appear and then instruct the system to create a data surface of icons for selected tuples in the database management system.

2.2.1 Icon Class Description

The SDMS provides the database administrator with a tool for describing the appearance of each icon as a function of attributes of tuples in the DBMS. That tool is the Icon Class Description Language (ICDL). It consists of a series of statements, each of which accept some attribute value and perform some graphical operation. The ICDL which was used to generate the icons shown in the preceding example is given in Figure 7.

```
icon class cluster(r)
begin
maximum size is (110,60);
position is
   (case r.type begin
                "CV"     :800
                "SSN"    :1600
                "SSBN"   :1600
                "SSGN"   :1600
                "CGN"    :2500
                "CG"     :2500
                "CA"     :2500
                "DDG"    :3200
                "FF"     :3200
                "AGI"    :3200
                "AO"     :4000
                default  :1600
                end,      case r.nat begin
                          "US":1200
                          "UR":2000
                          default:2500
                          end );

        template icon  case r.type  begin
                                    "CV"     :1
                                    "SSN"    :2
                                    "SSBN"   :2
                                    "SSGN"   :2
                                    "CGN"    :3
                                    "CG"     :3
                                    "CA"     :3
                                    "DDG"    :4
                                    "FF"     :4
                                    "AGI"    :4
                                    "AO"     :5
                                    default  :2
                                    end;
        scale is r.beam*2 percent;
        color of region 1 is
                   case r.ready begin
                                "1":green
                                "2":yellow
                                "3":orange
                                "4":red
                                default: gray
                                end ;
        attribute region r.nam from ( 5,16) to (70,28);
        attribute region r.ircs from ( 5,28) to (70,40);
end;
```

Figure 7: Icon Class Description

The position statement determines the placement of the icon on the data surface. In the example, it maps the ship's type into x-coordinates and nationality into y-coordinates.

The template statement specifies the shape of the icon by selecting among a set of pictures which have previously been drawn by the database administrator.

The scale statement specifies the size of the icon as a function of the beam of the ship.

The color statement specifies the color of each ship according to its readiness. Finally, the two attribute region statements place the values of the ship's name and international radio call sign into the specified locations in the icon.

2.2.2 The Association Statement

Having given the rules for the appearance of each icon, the database administrator creates the graphical representation using the associate statement. This causes SDMS to select particular tuples from a specified relation in the database and to create icons from them based on the designated icon class description.

To generate the graphical data surface of the preceding example, the database administrator would type:

```
associate ship using cluster
```

This causes SDMS to retrieve the tuples from the relation SHIP and pass them one at a time to a module which interprets the ICDL "cluster", using the values of the attributes of the tuple. For each such tuple, an icon is created on the graphical data surface.

The associate statement also permits the use of a qualification to select tuples. For example, a graphical data surface containing icons for those ships having a readiness of other than 1 could be created by typing:

```
associate ship using cluster where ship.ready != 1
```

This allows the combination of the capabilities of symbolic and graphic retrieval in searching for information.

3. CONTRASTS TO CONVENTIONAL DBMS

As can be seen from the above example, a Spatial Data Management System offers several advantages over conventional, keyboard-oriented database management systems, including those offering natural language or "English-like" user interfaces. This section discusses six such advantages:

1. Motion through the database is simple and natural.
2. The database is its own data dictionary.
3. The presentation of the data encourages browsing.
4. The placement of the data can convey information.
5. Graphics can be used to convey information.
6. The system can accommodate many unique data types such as photographs.

3.1 Motion Controls

The joy stick of SDMS provides a simple and natural means of moving through the database. By using one control, the joy stick, the user can explore the entire database. On the other hand, symbolic query languages, such as QUEL [YOUSSEFI, et al.], require the use of a special syntax and semantics. Even in natural language systems, where

the syntax is widely known and the semantics relatively intuitive, the user must learn the structure and contents of the database before he can find anything. In contrast, the data in an SDMS can be displayed in a manner which makes the contents and structure readily apparent, and does not require any prior knowledge of the structure of the database in order to retrieve information from it.

3.2 The Graphical Data Space as its own Data Dictionary

A conventional DBMS requires the use of a data dictionary to inform the user of the structure of the database. Even natural language user interfaces suffer from the problem of educating the user as to what queries may be answered from the information contained in the database. In contrast, the graphical data space of SDMS is its own data description. Rather than specify a relation and attribute name, the user merely traverses the data surface until he reaches the desired information, at which point the data is laid out in front of him.

Data types are shown by example. As there will be many such examples displayed on the data surface, they serve not only to display the data type which is permitted, but they also reveal the values of the data which are typi-

cally used. This property informs the user of the ranges of the data or the shades of meaning often obscured by attribute names such as "remarks" or "comments". Since the values of such attributes are displayed on the screen, the uses made of such fields may be readily ascertained.

3.3 Browsing

The user of an SDMS is almost always presented with a display which gives him more information than he immediately needs. Within this presentation, finding the required information is facilitated by the distinctive visual qualities which can be imparted to the data. At the same time, the "unsolicited" surrounding data makes it possible for the user to browse through the database. This is a difficult activity in a conventional database system where every piece of data must be requested explicitly. While a small database may be printed out and examined, the lack of any mechanism for placing related data together would make such a technique impractical for very large databases. Likewise, it would be very tedious to submit repeated queries and such a technique would be ineffective if the user was not already familiar with the contents and structure of the database.

In contrast, SDMS allows the database administrator to arrange the information in the database according to any chosen attributes. Once the user has positioned his window in the vicinity of the data being sought, he can browse through the surrounding area, letting the appearance of the icons determine where he focuses his attention.

3.4 Using Icon Position to Convey Information

The placement of a particular icon can be used to aid in recalling a particular datum, in much the same way that a person finds some needed information in his office by recalling where he put the piece of paper on which it was written.

The placement of an icon may also convey information directly. In the example of the preceding section, the location of each ship indicated its nationality and type. A personnel database could be arranged according to seniority or salary. The user could then observe the world-view map to get an overall picture, such as seeing how many items were in each category, and he could move his magnified window over some particular area to look at exceptional, extreme, or average values.

3.5 Graphic Representations

Graphic representations are often the most vivid means of conveying information, especially for aiding in the perception of trends in large aggregations of data. The most familiar form of this technique is the histogram or the graph, where numeric data is displayed in graphical form. The graphical output facilites of SDMS make such display of numeric data possible as a natural extension of the system.

Graphical representations may also be used to advantage in *displaying trends in non-numeric* data. For example, a database of ships could be displayed against the background of a map, with the wake behind each ship indicating its speed and direction. With such a display, it would be easy to spot trends that would be hard to formulate into symbolic queries, such as a large number of ships heading for the Middle East.

3.6 New Data Types

An SDMS is not restricted to data that originates as numbers and character strings. The raster scan output devices and digital storage of graphical data provide a natural means of storing images such as photographs. The prototype implementation at CCA also includes an optical videodisk which can be controlled through SDMS to provide for storing a very large number of video images.

4. OPTICAL VIDEODISK

The CCA SDMS prototype incorporates an optical analogue video disk player. Optical videodisks have a capacity of 54,000 images, which can viewed either as discrete photographs or in sequence as a motion picture. The same number of images stored on a conventional magnetic digital disk would require 10**11 bits, or 100 3330-type disks.

The spatial, graphical access to information of SDMS provides a natural mechanism for retrieving spatial or graphical information stored on the optical videodisk. Two approaches are employed for accessing video disk images.

Data which is inherently spatial, such as a map, is divided into sections small enough to be stored as individual video frames. The user selects a particular frame through the use of a digitally-stored map, as shown in figure 8. He can traverse this data surface as he would any other graphical data surface. When he zooms in to get a closer look, the system displays the videodisk image from the corresponding position.

A second method of accessing videodisk data is used for accessing photographs which are not spatially related, as would be the case with photographs of employees in a personnel database. In this case, illustrated in figure

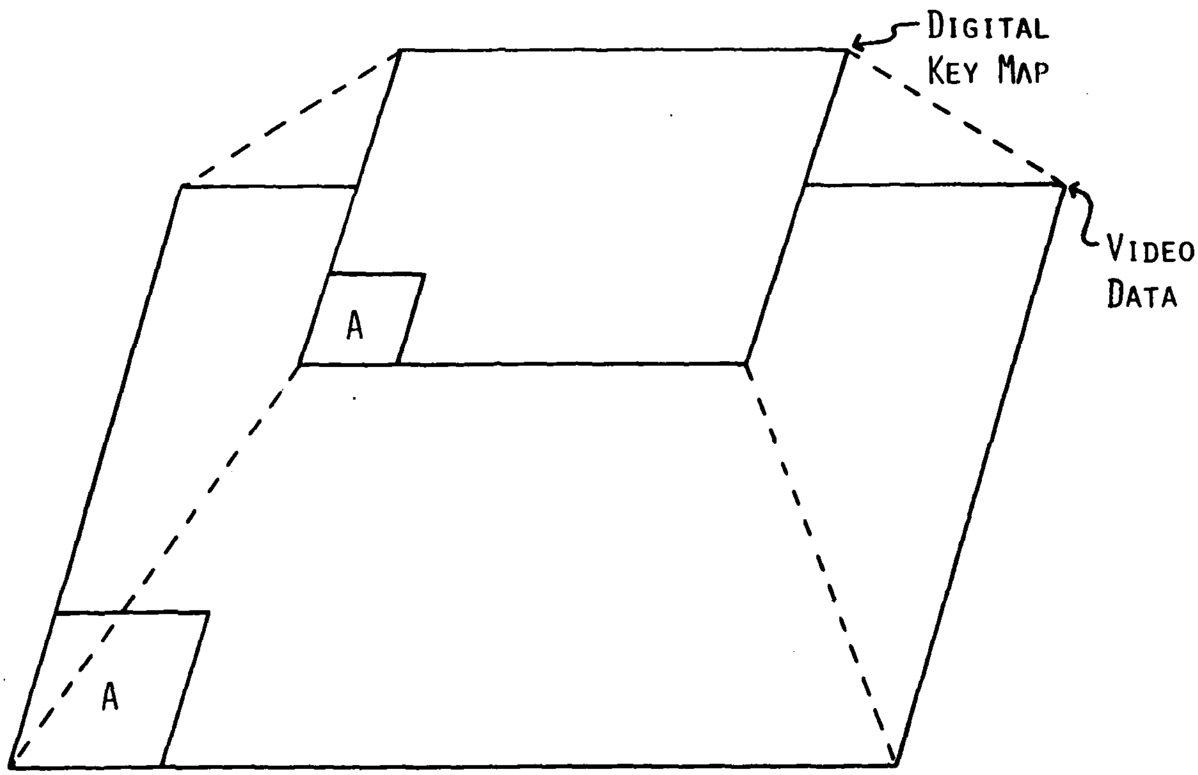


FIGURE 8: SPATIAL DATA

9, the user once again begins with a digitally-stored graphical data surface. This surface is populated with discrete icons, one for each frame to be accessed. By zooming in on a particular icon, the user can see the associated frame. A variation on this scheme allows one icon to be associated with a sequence of videodisk images. When the icon is selected, that sequence may be examined as a motion picture or as individual photographs, with the user specifying the location in the sequence where he wishes to begin.

The mixture of analogue and digital video in SDMS also makes possible the editing of data which is stored on the read-only video disk. As the output of the videodisk can be directed to the center display screen at the same time that it is displaying digitally-stored information, the two can be superimposed, providing a means to overlay analogue video with digital annotations. Alternatively, the analogue video may be digitized and operated upon as bit array data.

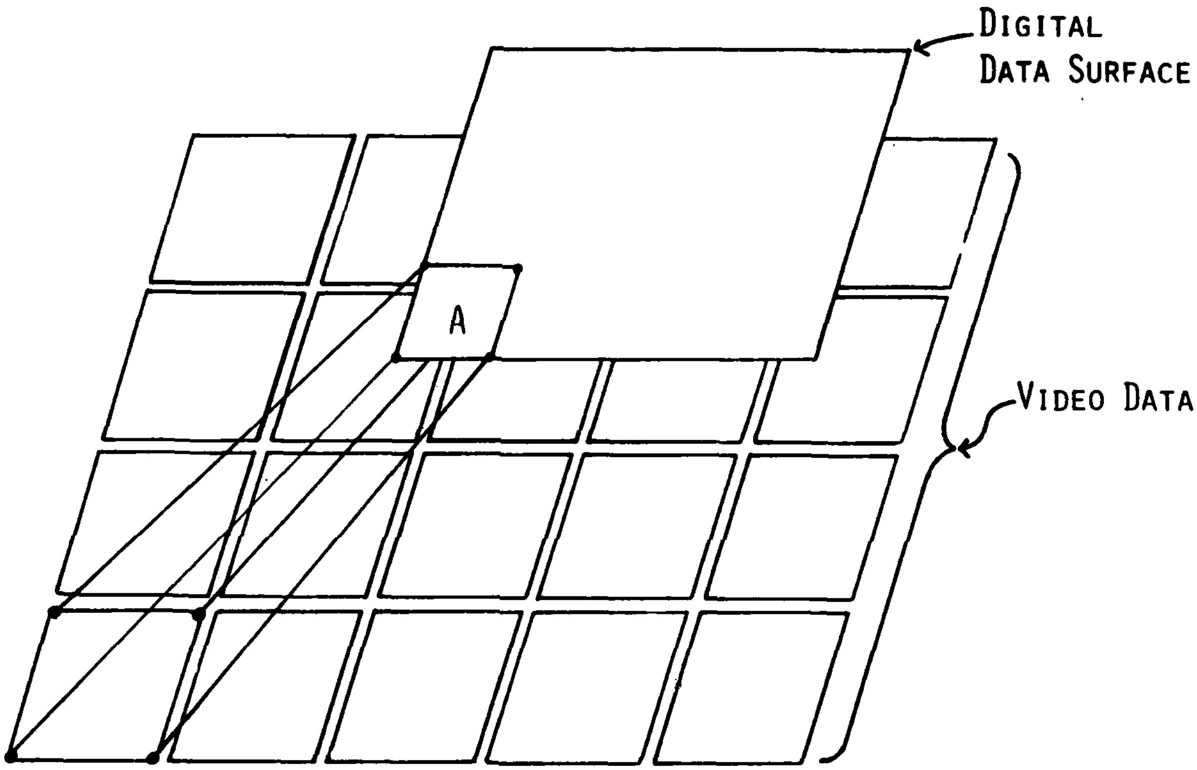


FIGURE 9: PHOTOGRAPHIC DATA

5. DEFINING GRAPHICAL VIEWS

This section provides additional details on the relationship between symbolic and graphical representations of data in SDMS. It introduces several features not mentioned in the example of Section 2.

5.1 Dual Representation of Data

5.1.1 Entities and Icons

The SDMS allows the tuples of a relation in a symbolic database management system (DBMS) to be represented in a graphical data space (GDS).* The graphic representation of a tuple is an icon. Those tuples having a graphic representation are distinguished from other tuples and are called entities. Each entity may be represented by more than one icon. Conversely, each icon may be represented by more than one entity.

---

*In the CCA implementation of SDMS, the DBMS used is INGRES, a relational database system [HELD, STONEBRAKER, WONG]. The query language of INGRES is called QUEL.

A goal of SDMS is to allow a user to refer to data via graphic methods such as spatial search or via symbolic methods, such as using a QUEL retrieval statement. To achieve this, SDMS provides a mapping between each entity and its icon. This mapping is provided via a link which logically connects the two. When an entity and icon are linked, a selection of one implies a selection of the other. In the query language of SDMS, the two selection methods may be combined through the use of links.

Links are created by associating a tuple with an icon, which makes the tuple an entity. There are two types of associations in SDMS. The first type, the specific association links a specific tuple to an existing icon. It is most useful for recording symbolic information about visual data, such as entering a description of the subject matter of a photograph.

The second type, the class association, generates new icons for one or more tuples in a relation. The class association was used to create the graphical view of the ship's database used in Section 2.

Specific associations are most useful for entering symbolic data for use in locating information that is inherently graphical, while class associations are most useful for creating graphical data for use in finding symbolic data which is inherently symbolic. The class

association eliminates the need for explicitly creating and linking an icon to each tuple by providing an automated mechanism for doing so.

5.1.2 Specific Associations

A specific association between a tuple and an icon creates a link between the two. The tuple must already exist in some relation at the time of the association. The icon must also exist in the GDS at the time of the association. Such icons will often be photographs or manually constructed diagrams. By creating tuples and linking them to such an icon, the user can then locate specific icons by the use of symbolic queries. For example, a database of paintings could be accessed by means of a work's artist, year, nationality, genre, etc.

Specific associations can also be used in conjunction with sub-icons (described in Section 5.5) to allow a class association to specify individual photographs. For example, the icons for the employees in a personnel database could each include a photograph of that employee.

5.1.3 Class Associations

The class association is the principle tool for connecting the Graphical Data Space (GDS) to the DBMS. It is intended primarily as a tool for the database administrator (DBA).

A class association between a relation and the GDS causes the creation of icons which graphically represent the tuples of the relation. The icons are placed in the GDS. It is possible to associate only a subset of the relation by supplying a qualification which determines which tuples are to be represented. It is also possible to have the placement of the new icons restricted to a particular rectangular region of the GDS. The effects of such an association are:

1. For each tuple in the relation, an icon is created and inserted into the GDS. If a qualification was supplied, icons are created for only those tuples which pass the qualification.

2. Each of these tuples and their corresponding icons are linked. The system will maintain the correspondence between the two as the database is updated.

When a class association is made, an icon class description may be specified which describes exactly how to draw each icon. The icon class is essentially a picture where

certain parameters may vary each time the picture is drawn. These parameters include size, shape, color, position and text strings. The values for these parameters may depend on the data in the entity being represented. Icon classes are discussed in Section 5.2.

When a class association is made, an explicit data dependence is established between each entity and its corresponding icon. This is manifested at first when the system draws these icons such that they reflect the data in the entity. Secondly, the data dependence serves to insure that the icons which are created due to a class association will always reflect the tuples they represent, until the association is explicitly broken. Hence, if an entity is updated, its corresponding icon is updated. If an entity is deleted, its icon is erased. If new tuples are added to the relation, they become entities if they pass the qualification and new icons are created to represent them. The result is that the GDS always contains a representation for the given relation. The GDS serves as an alternate way to view the relation.

5.2 Icon Classes

An icon class is used in conjunction with a class association. While specific and class associations are relatively simple and may be performed by ordinary SDMS users, the icon class description language (ICDL) is more complex and is meant to be utilized primarily by the database administrator.

An icon class description consists of a series of statements which specify the appearance of the icon. Each statement performs some graphical operation, such as selecting a template picture, coloring some region, or inserting some text. The values of arguments to an ICDL statement may be taken from attributes of tuples retrieved from INGRES.

An icon which is constructed from an icon class may have its appearance defined at several levels of detail. This allows the "zooming" effect as illustrated in Section 2, by which a user sees a more detailed version of an icon by magnifying it. The bit-array description for a single level of detail is referred to as an <u>image plane</u>. The pictures for each image plane originate as <u>templates</u>, which are simply pictures drawn by the database administrator on a special image plane reserved for that purpose. For example, the templates for ships might be drawn as

follows: At the top-most (least detail) image plane, the ship appears as a small rectangle. The second image plane has a rough silhouette with the ship's name and radio call sign beneath it. The third (most detailed) image plane is a more detailed pictur showing some of the ship's superstructure. The ship's name, call sign, beam, draft, speed, etc., appear beneath the picture. With such a description, a user who "zooms" in on such an icon will get more detail as he gets closer.

Continuing with this example, we may want different drawings for different types of ships. An icon class may include a rule for selecting among several templates.

The parameters of an icon which may be controlled by an icon class description are:

1. maximum size: The maximum size of each icon may be specified to prevent any one of them from becoming too large. Since relative size is a picture parameter, it is a good protection.

2. choice of picture: There may be several different pictures which serve as templates, depending upon the data in the tuple. In the example in Section 2 carriers had a different picture from destroyers. If more than one picture is given, a rule for deciding which picture to use must be provided.

3. target position: The target position specifies the exact location for an icon within the GDS. SDMS does not let icons overlap, so if an icon can fit at the target position, it will be placed there. Otherwise, a location close to the target position is used.

4. color: Any region of the template may be colored. In the example, the readiness determined the color of each ship.

5. size: The template given for the description of an icon class may be scaled. In the example, the size was a function of the beam of each ship. The default size is the size of the original template, up to the maximum size for the icon class.

6. orientation: The picture may be oriented by some arbitrary angle. The default is the orientation of the original template.

7. text: Text may be added to the picture. This text may be a simple string or it could be the data from one or more fields of the tuple.

5.3 Using INGRES data

Each icon in the GDS may reflect the data in the entity it represents. This is done by making the data in the entity accessible from within the icon class description. It is also possible to retrieve other tuples from any relation in INGRES to aid in the description of an icon class. There are two statements to perform retrievals. One method is via the get statement. The get statement is used when one tuple is needed in response to a query, as in finding the beam of a particular ship. If more than one tuple satisfies the qualification of the statement, an error occurs. A second mechanism retrieves *a set of tuples* from a relation, by means of a for loop statement. This statement uses one or more statements which are executed once for each tuple retrieved. The for loop construct could be used to retrieve the previous positions of a ship from a track history relation. Each position could then be displayed as part of the wake of the ship.

5.4 Sub-icons

Another feature of an icon class is the ability to have sub-icons drawn within the area of an icon. This feature allows the nesting of icons, for example displaying a sub-icon for each employee within a department that is represented by one icon. Sub-icons may be included to overlay portions of the picture for some or all levels of detail.

The sub-icon differs from a normal icon in that:

1. If the "parent" icon is moved, the sub-icons move with it.
2. If the "parent" icon is erased, and its link broken, all of its sub-icons are erased and their links are broken.

5.5 SQUEL - The Query Language of SDMS

The query language of SDMS is a combination of QUEL, the query language of INGRES, plus additions made for the graphical environment of SDMS. The name SQUEL arises from Spatial QUEry Language. Statements from QUEL can be entered directly to SQUEL, as QUEL is a proper subset.

Several commands are added to the original QUEL commands to produce SQUEL.

1. <u>blink</u> <relation> <u>where</u> <qual>

   Blink finds all the tuples which satisfy <qual>. Any icons in the current I-Space which correspond to those tuples will blink. They continue to blink until a null <u>blink</u> command is entered or until the user exits the system.

2. <u>frame</u> <relation> <u>where</u> <qual>

   Similar to <u>blink</u> except that the appropriate icons are framed, not blinked. Framing an icon simply draws a rectangle around it. Framing is erased similarly to blinking.

3. <u>associate</u> <relation> using <icdl> where <qual>

   creates an icon for each tuple which meets the qualification. The appearance of each icon is determined by the given icdl.

4. <u>find</u> <tuple var> <u>where</u> <qual>

   Used for a rapid transit to the location of the icon corresponding to the tuple retrieved. If more than one tuple is retrieved, or if it is represented in more than one place in the GDS, the user is asked to decide upon which one to go to. Confirmation is required before the move actually occurs.

5. <u>point</u>

   Informs SDMS that the user will point to an icon. When he does, the data in the tuple corresponding to the selected icon is displayed on the graphics screen, or on the text display.

6. <u>change</u>

   Informs SDMS that the user will point to a position within an icon. This signals that the user will update the attribute displayed there. The user then enters the new value for the attribute and the update is made immediately. This allows the user to update the database without requiring any symbolic specification of the target of the updating operation.

6. SDMS PROTOTYPE IMPLEMENTATION

This section describes the particular prototype of SDMS implemented at Computer Corporation of America. An earlier implementation of SDMS which utilized manually entered images, as opposed to data derived from a symbolic database, was built at the Architecture Machine Group at the Massachusetts Institute of Technology and is described in [DONELSON] and [BOLT].

6.1 System Environment

The prototype SDMS is written in the C language, running under the Unix operating system on a DEC PDP11/70. The machine has 1 megabyte of primary memory and 176 megabytes of moving-head disk storage. Much of the memory is used for storage and manipulation of the bit-map representations of icons and graphical data spaces.

The graphical data space is viewed via a Lexidata frame buffer display. This display has its own 480x640x8 bit memory from which the image on the color television screen is generated.

6.2 Schedule

The SDMS prototype implementation effort has been underway for a little over a year. The current version of the system provides all of the facilities described in Section 2 for moving about the graphical data surface and generating graphical views of symbolic data. Some of the advanced features described in Section 5, such as sub-icons, *set*, *for-loop*, *blink*, *frame*, and *change* are planned for implementation by this coming fall.

The current implementation consists of approximately 40,000 lines of code.

References

[BOLT]
Richard Bolt Spatial Data Management, DARPA Report Massachusetts Institute of Technology Architecture Machine Group, Cambridge, Massachusetts, 1979.

[DONELSON]
William C. Donelson, Spatial Management of Information, SIGGRAPH 78, Atlanta, Ga., 1978.

[HELD STONEBRAKER WONG]
Stonebraker, M.R.; Held, G.D.; Wong, E. "INGRES - A relational data base system", AFIPS Proceedings, Volume 44.

[YOUSSEFI et al.]
Ingres Reference Manual, Memorandum No. ERL-M579, University of California, Berkeley, 1977.